AMENDMENTS TO THE CLAIMS

The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method of processing an undeniable digital signature, comprising the steps of:

(a) generating public keys (D, P, k, t) and secret keys (D1, q) at a signer side, by generating two primes p, q (p, q > 4, p = 3 mod 4, $\sqrt{\frac{p}{3}} < q$), computing D1 = -p and D = $D1q^2$, obtaining a bit length k of $\frac{\sqrt{|D1|}}{4}$ and a bit length t of q-(D1/q) where (D1/q) denotes Kronecker symbol, and generating a kernel element P of a map from a class group Cl(D) to a class group Cl(D1);

(b) generating a signature S for a message m at the signer side, by embedding the message m into a message ideal M in the class group Cl(D) where a norm of the message ideal M is larger than k+1 bits, and mapping the message ideal M to the class group Cl(D1) and pulling the mapped message ideal M back to the class group Cl(D); and

(c) verifying the signature S by:

(c1) checking whether a norm N(S) of the signature S received from the signer side is smaller than k bits or not, and judging that the signature S is illegal when the norm N(S) is larger than k bits, or generating a challenge C when the norm N(S) is not larger than k bits, by computing the message ideal M of the message m, generating a random integer r smaller than t bits, computing $H = (M/S)^r$, generating a random ideal B whose norm is smaller than k-1 bits, and computing the challenge C = BH, at a verifier side;

(c2) computing a response W by mapping the challenge C received from the verifier side to the class group Cl(D1) and pulling the mapped challenge C back to the class

group Cl(D) and squaring a result of mapping and pulling back, using the secret keys (D1, q), at the signer side; and

(c3) checking whether $W = B^2$ holds or not by using the response W received from the signer side, and judging that the signature S is legal when $W = B^2$ holds or that the signature S is illegal otherwise, at the verifier side.

2. (Original) A signer device for processing an undeniable digital signature, comprising:

a key generation unit for generating public keys (D, P, k, t) and secret keys (D1, q), by generating two primes p, q (p, q > 4, p = 3 mod 4, $\sqrt{\frac{p}{3}} < q$), computing D1 = -p and D = $D1q^2$, obtaining a bit length k of $\frac{\sqrt{|D1|}}{4}$ and a bit length t of q-(D1/q) where (D1/q) denotes Kronecker symbol, and generating a kernel element P of a map from a class group Cl(D) to a class group Cl(D1);

a signature generation unit for generating a signature S for a message m, by embedding the message m into a message ideal M in the class group Cl(D) where a norm of the message ideal M is larger than k+1 bits, and mapping the message ideal M to the class group Cl(D1) and pulling the mapped message ideal M back to the class group Cl(D); and

a response generation unit for receiving a challenge C = BH from a verifier side, where B is a random ideal whose norm is smaller than k-1 bits, $H = (M/S)^r$, and r is a random integer smaller than t bits, computing a response W by mapping the challenge C to the class group Cl(D1) and pulling the mapped challenge C back to the class group Cl(D) and squaring a result of mapping and pulling back, using the secret keys (D1, q), and sending the response W to the verifier side, in a process for verifying the signature S.

3. (Currently amended) A verifier device for processing an undeniable digital signature, using a message m and a signature S for the message m received from a signer

side, where public keys (D, P, k, t) and secret keys (D1, q) are defined by generating two primes p, q (p, q > 4, p = 3 mod 4, $\sqrt{\frac{p}{3}} < q$) ,computing D1 = -p and D = D1q^2, obtaining a bit length k of $\frac{\sqrt{|D1|}}{4}$ and a bit length t of q-(D1/q) where (D1/q) denotes Kronecker symbol, and generating a kernel element P of a map from a class group Cl(D) to a class group Cl(D1), and the signature S ~~for the message m~~ is generated by embedding the message m into a message ideal M in the class group Cl(D) where a norm of the message ideal M is larger than k+1 bits, and mapping the message ideal M to the class group Cl(D1) and pulling the mapped message ideal M back to the class group Cl(D), the verifier device comprising:

a norm checking unit for checking whether a norm N(S) of the signature S is smaller than k bits or not, and judging that the signature S is illegal when the norm N(S) is larger than k bits;

a challenge generation unit for generating a challenge C when the norm N(S) is not larger than k bits, by computing the message ideal M of the message m, generating a random integer r smaller than t bits, computing H = $(M/S)^r$, generating a random ideal B whose norm is smaller than k-1 bits, and computing a challenge C = BH, and for sending the challenge C to ~~a~~ the signer side; and

a response checking unit for receiving a response W from the signer side, checking whether W = B^2 holds or not, and judging that the signature S is legal when W = B^2 holds or that the signature S is illegal otherwise, where the response W being obtained by mapping the challenge C to the class group Cl(D1) and pulling the mapped challenge C back to the class group Cl(D) and squaring a result of mapping and pulling back, using the secret keys (D1, q).

4. (Original) A computer usable medium having computer readable program codes embodied therein for causing a computer to function as a signer device for processing an undeniable digital signature, the computer readable program codes including:

a first computer readable program code for causing said computer to generate public keys (D, P, k, t) and secret keys (D1, q), by generating two primes p, q (p, q > 4, p = 3 mod 4, $\sqrt{\frac{p}{3}} < q$), computing D1 = -p and D = D1q^2, obtaining a bit length k of $\frac{\sqrt{|D1|}}{4}$ and a bit length t of q-(D1/q) where (D1/q) denotes Kronecker symbol, and generating a kernel element P of a map from a class group Cl(D) to a class group Cl(D1);

a second computer readable program code for causing said computer to generate a signature S for a message m, by embedding the message m into a message ideal M in the class group Cl(D) where a norm of the message ideal M is larger than k+1 bits, and mapping the message ideal M to the class group Cl(D1) and pulling the mapped message ideal M back to the class group Cl(D); and

a third computer readable program code for causing said computer to receive a challenge C = BH from a verifier side, where B is a random ideal whose norm is smaller than k-1 bits, H = $(M/S)^r$, and r is a random integer smaller than t bits, compute a response W by mapping the challenge C to the class group Cl(D1) and pulling the mapped challenge C back to the class group Cl(D) and squaring a result of mapping and pulling back, using the secret keys (D1, q), and send the response W to the verifier side, in a process for verifying the signature S.

5. (Currently amended) A computer usable medium having computer readable program codes embodied therein for causing a computer to function as a verifier device for processing an undeniable digital signature, using a message m and a signature S received from a signer side, where public keys (D, P, k, t) and secret keys (D1, q) are defined by generating two primes p, q (p, q > 4, p = 3 mod 4, $\sqrt{\frac{p}{3}} < q$), computing D1 = -p and D = D1q^2, obtaining a bit length k of $\frac{\sqrt{|D1|}}{4}$ and a bit length t of q-(D1/q) where (D1/q) denotes

Kronecker symbol, and generating a kernel element P of a map from a class group Cl(D) to a class group Cl(D1), and the signature S for the message m is generated by embedding the message m into a message ideal M in the class group Cl(D) where a norm of the message ideal M is larger than k+1 bits, and mapping the message ideal M to the class group Cl(D1) and pulling the mapped message ideal M back to the class group Cl(D), the computer readable program codes including:

a first computer readable program code for causing said computer to check whether a norm N(S) of the signature S is smaller than k bits or not, and judge that the signature S is illegal when the norm N(S) is larger than k bits;

a second computer readable program code for causing said computer to generate a challenge C when the norm N(S) is not larger than k bits, by computing the message ideal M of the message m, generating a random integer r smaller than t bits, computing $H = (M/S)^r$, generating a random ideal B whose norm is smaller than k-1 bits, and computing the challenge $C = BH$, and send the challenge C to ~~a~~ the signer side; and

a third computer readable program code for causing said computer to receive a response W from the signer side, check whether $W = B^2$ holds or not, and judge that the signature S is legal when $W = B^2$ holds or that the signature S is illegal otherwise, where the response W being obtained by mapping the challenge C to the class group Cl(D1) and pulling the mapped challenge C back to the class group Cl(D) and squaring a result of mapping and pulling back, using the secret keys (D1, q).

6.-7. (Cancelled).

8. (Currently amended) ~~The method of claim 7~~ A method for providing a software vending service, comprising the steps of:

(a) attaching a signature S to a software offered for downloading by clients at a software vendor side, according to an undeniable digital signature scheme, wherein the step (a) further includes the steps of:

(a1) generating public keys (D, P, k, t) and secret keys (D1, q) at the software vendor side, by generating two primes p, q (p, q > 4, p = 3 mod 4, $\sqrt{\frac{p}{3}} < q$), computing D1 = -p and D = D1q^2, obtaining a bit length k of $\frac{\sqrt{|D1|}}{4}$ and a bit length t of q-(D1/q) where (D1/q) denotes Kronecker symbol, and generating a kernel element P of a map from a class group Cl(D) to a class group Cl(D1); and

(a2) generating the signature S for a message m representing the software at the software vendor side, by embedding the message m into a message ideal M in the class group Cl(D) where a norm of the message ideal M is larger than k+1 bits, and mapping the message ideal M to the class group Cl(D1) and pulling the mapped message ideal M back to the class group Cl(D); and

(b) verifying the signature S a client side which has downloaded the software with the signature S attached thereto interactively with the software vendor side, so as to prove that the software has not been altered from an original, wherein the step (b) further includes the steps of:

(b1) checking whether a norm N(S) of the signature S received from the software vendor side is smaller than k bits or not, and judging that the signature S is illegal when the norm N(S) is larger than k bits, or generating a challenge C when the norm N(S) is not larger than k bits, by computing the message ideal M of the message m, generating a random integer r smaller than t bits, computing H = (M/S)^r, generating a random ideal B whose norm is smaller than k-1 bits, and computing the challenge C = BH, at a the client side;

(b2) computing a response W by mapping the challenge C received from the client side to the class group Cl(D1) and pulling the mapped challenge C back to the class group Cl(D) and squaring a result of mapping and pulling back, using the secret keys (D1, q), at the software vendor side; and

(b3) checking whether $W = B^2$ holds or not by using the response W received from the software vendor side, and judging that the signature S is legal when $W = B^2$ holds or that the signature S is illegal otherwise, at the client side.

9. (Currently amended) The method of claim ~~6~~ 8, wherein the step (a) attaches the ~~undeniable digital~~ signature S using different sets of public keys and secret keys for different ~~softwares~~ kinds of software.

10.-12. (Cancelled).

13. (Currently amended) ~~The method of claim 12~~ A method for enabling a user side to check authenticity of an e-commerce/information service provider, comprising the steps of:

(a) obtaining public keys (D, P, k, t), secret keys (D1, q), and a signature S for the public keys from a certificate authority side at the e-commerce/information service provider, the signature being generated by the certificate authority side according to an undeniable digital signature scheme, wherein the step (a) further includes the steps of:

(a1) generating the public keys and the secret keys at the certificate authority side, by generating two primes p, q ($p, q > 4$, $p = 3 \bmod 4$, $\sqrt{\frac{p}{3}} < q$), computing $D1 = -p$ and $D = D1q^2$, obtaining a bit length k of $\frac{\sqrt{|D1|}}{4}$ and a bit length t of q-(D1/q) where (D1/q) denotes Kronecker symbol, and generating a kernel element P of a map from a class group Cl(D) to a class group Cl(D1); and

(a2) generating the signature S for the public keys at the certificate authority side, by embedding the public keys into a message ideal M in the class group Cl(D) where a

norm of the message ideal M is larger than k+1 bits, and mapping the message ideal M to the class group Cl(D1) and pulling the mapped message ideal M back to the class group Cl(D);

(b) providing the public keys and the signature S from the e-commerce/information service provider to the user side, such that the user side carries out a process of verifying the signature S provided from the e-commerce/information service provider to the user side, interactively with the certificate authority side to prove authenticity of the public keys provided by the e-commerce/information service provider, wherein ~~at~~ the step (b) ~~the signature is verified by~~ further includes the steps of:

(b1) checking whether a norm N(S) of the signature S received from the certificate authority side is smaller than k bits or not, and judging that the signature S is illegal when the norm N(S) is larger than k bits, or generating a challenge C when the norm N(S) is not larger than k bits, by computing the message ideal M of the public keys, generating a random integer r smaller than t bits, computing $H = (M/S)^r$, generating a random ideal B whose norm is smaller than k-1 bits, and computing the challenge $C = BH$, at ~~a~~ the user side;

(b2) computing a response W by mapping the challenge C received from the user side to the class group Cl(D1) and pulling the mapped challenge C back to the class group Cl(D) and squaring a result of mapping and pulling back, using the secret keys (D1, q), at ~~a~~ the certificate authority side; and

(b3) checking whether $W = B^2$ holds or not by using the response W received from the certificate authority side, and judging that the signature S is legal when $W = B^2$ holds or that the signature S is illegal otherwise, at the user side; and

(c) receiving an encrypted random data from the user side, the encrypted random data being encrypted by the user using the public keys, decrypting the encrypted random data using the secret keys, and returning a decrypted random data to the user side, such that the user side checks if the decrypted random data coincides with an original random data to prove that the e-commerce/information service provider has authentic secret keys.

14.-16. (Cancelled).

17. (Currently amended) ~~The method of claim 16~~ A method for enabling a user side to check authenticity of an e-commerce/information service provider, comprising the steps of:

(a) issuing public keys (D, P, k, t), secret keys (D1, q), and a signature S for the public keys from a certificate authority side to the e-commerce/information service provider, the signature S being generated according to an undeniable digital signature scheme, wherein the step (a) further includes the steps of:

(a1) generating the public keys and the secret keys at the certificate authority side, by generating two primes p, q (p, q > 4, p = 3 mod 4, $\sqrt{\frac{p}{3}} < q$, computing D1 = -p and D = D1q^2, obtaining a bit length k of $\frac{\sqrt{|D1|}}{4}$ and a bit length t of q-(D1/q) where (D1/q) denotes Kronecker symbol, and generating a kernel element P of a map from a class group Cl(D) to a class group Cl(D1); and

(a2) generating the signature S for the public keys at the certificate authority side, by embedding the public keys into a message ideal M in the class group Cl(D) where a norm of the message ideal M is larger than k+1 bits, and mapping the message ideal M to the class group Cl(D1) and pulling the mapped message ideal M back to the class group Cl(D); and

(b) verifying the signature S provided from the e-commerce/information service provider to the user side, at the certificate authority side interactively with the user side in order to prove authenticity of the public keys provided by the e-commerce/information service provider, wherein ~~at~~ the step (b) ~~the signature is verified by~~ further includes the steps of:

(b1) checking whether a norm N(S) of the signature S received from the certificate authority side is smaller than k bits or not, and judging that the signature S is illegal when the norm N(S) is larger than k bits, or generating a challenge C when the norm N(S) is not larger than k bits, by computing the message ideal M of the public keys, generating a random integer r smaller than t bits, computing $H = (M/S)^r$, generating a random ideal B whose norm is smaller than k-1 bits, and computing the challenge C = BH, at a the user side;

(b2) computing a response W by mapping the challenge C received from the user side to the class group Cl(D1) and pulling the mapped challenge C back to the class group Cl(D) and squaring a result of mapping and pulling back, using the secret keys (D1, q), at a the certificate authority side; and

(b3) checking whether $W = B^2$ holds or not by using the response W received from the certificate authority side, and judging that the signature S is legal when $W = B^2$ holds or that the signature S is illegal otherwise, at the user side.

18.-20. (Cancelled).

21. (Currently amended) ~~The method of claim 20~~ A method for enabling a user side to check authenticity of an e-commerce/information service provider, comprising the steps of:

(a) generating a signature S for a hash value of a home page of the e-commerce/information service provider at a certificate authority according to an undeniable digital signature scheme, wherein the step (a) further includes the steps of:

(a1) generating public keys (D, P, k, t) and secret keys (D1, q) at the certificate authority, by generating two primes p, q (p, q > 4, p = 3 mod 4 $\sqrt{\frac{p}{3}}$ <), computing

D1 = -p and D = D1q^2, obtaining a bit length k $\frac{\sqrt{|D1|}}{4}$ and a bit length t of q-(D1/q) where (D1/q) denotes Kronecker symbol, and generating a kernel element P of a map from a class group Cl(D) to a class group Cl(D1); and

(a2) generating the signature S for the hash value of the home page at the certificate authority, by embedding the hash value of the home page into a message ideal M in the class group Cl(D) where a norm of the message ideal M is larger than k+1 bits, and mapping the message ideal M to the class group Cl(D1) and pulling the mapped message ideal M back to the class group Cl(D);

(b) posting the signature on a display of the home page of the e-commerce/information service provider at a user side from the certificate authority side, such that the user side can initiate a process of verifying the signature by clicking the signature on the display; and

(c) verifying the signature S at the certificate authority side interactively with the user side in order to prove authenticity of the e-commerce/information service provider, wherein ~~at~~ the step (c) ~~the signature is verified by~~ further includes the steps of:

(c1) checking whether a norm N(S) of the signature S received from the certificate authority side is smaller than k bits or not, and judging that the signature S is illegal when the norm N(S) is larger than k bits, or generating a challenge C when the norm N(S) is not larger than k bits, by computing the message ideal M of the public keys, generating a random integer r smaller than t bits, computing H = (M/S)^r, generating a random ideal B whose norm is smaller than k-1 bits, and computing the challenge C = BH, at the user side;

(c2) computing a response W by mapping the challenge C received from the user side to the class group Cl(D1) and pulling the mapped challenge C back to the class group Cl(D) and squaring a result of mapping and pulling back, using the secret keys (D1, q), at a the certificate authority side; and

(c3) checking whether $W = B^2$ holds or not by using the response W received from the certificate authority side, and judging that the signature S is legal when $W = B^2$ holds or that the signature S is illegal otherwise, at the user side.